Stern Series Based ECC for Digital Signature with Message Recovery

Latha Parthiban[1], Nivethaa Sree[2]

[1]Department of Computer Science, Pondicherry University, Puducherry, India

[2]Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur, India

[*1]lathaparthiban@yahoo.com; [2]nivethaasree@yahoo.com

Abstract

A novel method of digital signature generation along with message recovery based on Elliptic Curve Cryptography (ECC) and knapsack algorithm is presented in this paper. In digital signature along with message recovery scheme, signature alone is sent and message will be recovered from the signature (r, s). ECC provides greater security with less key size, when compared to integer factorisation and discrete logarithm system. As the strength of knapsack algorithm depends on the selection of the series, the proposed algorithm uses modified Stern series which not only reduces the time complexity but also provides better security.

Keywords

Message Recovery; ECC; Knapsack Algorithm; Modified Stern Series

Introduction

Digital signature authenticates the identity of message sender to ensure that the message sent is unchanged. It also ensures message integrity, authentication and confidentiality the origin of a message. Digital signature with message recovery schemes reduces transmission costs as the message is contained in the signature itself and message and signature need not be sent separately. It is also suitable for key exchange applications, due to the small size of the key. In this paper, a new digital signature scheme along with message recovery using modified Stern series in knapsack algorithm has been proposed. Exponential series is used in existing method (Rajaram Ramasamy 2011) which has higher computational time when compared to modified Stern series proposed in our method.

The proposed method has four levels of authenticated encryption. The first level is based on elliptic curve signature, the second level is to apply knapsack value to the signing message, the third level applies to encryption using receiver's public key and the forth level repeats the last step using secret key generated based on elliptic curve Diffie-Hellman algorithm. The proposed method provides high security with reasonable computational cost. It is computationally infeasible for the intruders to find the private key from the publicly known domain parameters due to the difficulty in computing Elliptic curve discrete logarithm problem (ECDLP).

In this paper, section 2 deals with literature survey, section 3 is related to the proposed new digital signature scheme which applies knapsack on ECC with usage of modified Stern series in knapsack algorithm, section 4 shows experimental results that have difference in execution time between exponential and Stern series and in section 5 the concluding remarks are provided.

Related Work

In RSA (Rivest et al 1978), message was given as input to the hash function that produced the desired hash code encrypted using sender's private key to generate the signature. Message and signature were transmitted to the receiver and the sender's public key was used to decrypt the signature. Signature will be accepted as authenticated, if the calculated hash code and the decrypted signature are same. ElGamal (1985) proposed digital signature scheme based on the difficulty of computing discrete logarithms over a finite field with a large prime. A method to construct directly digital signature scheme is proposed by Noor Dhia Kadhm (2013).

National Institute of Standards and Technology (1992) published the Digital Signature Algorithm (DSA) in the Digital Signature Standard (DSS) to generate digital signature which cannot be used for encryption or key exchange, but RSA can be used for both encryption and digital signature. Nyberg and Rueppel's approach (1993) was based on the same principles as DSA along with the implementation of

message recovery. In Hsu and Wu scheme (1998), any ciphertext of a signature (sign first, then encrypt) for a message was sent to a specified group of verifiers, which follows the *(t, n)* threshold scheme in order to decrypt the ciphertext of signature. In (*t*, *n*) threshold scheme, any *t* out of *n* verifiers in the group shared the responsibility for message recovery.

ECC, introduced by Koblitz and Miller (1987) obtained its high level of security from the concept of the ECDLP in which it is difficult to determine *k* when *Q* and *P* are given which denotes the points in elliptic curve where $Q = kP$. ECC provides better level of security (Sonali and Malk, 2013) with less key size and higher computational efficiency, which can be used in resource - constrained device like cell phone, smart cards etc and for e-commerce applications (Asvhini et al 2012). ECC 160 bit provides same level of security when compared to RSA -1024 bit..

Chen et al (2002) combined one way hash function and the identification scheme by Popescu (2000) based on zero-knowledge, digital signature scheme. The design of one-way hash function is given with two characteristics: *O*ne is that the output will be of fixed length instead of the various length of input; the other is that the length of message with the signature can be reduced into a shorter digest through the hash function.

In Wu–Lin's (2008) approach, both authentication of the public key and verification of the signature can be done in one step with the concept of self certified public key cryptosystem. Convertible authenticated encryption based on ECC provides the computational secrecy, i.e., the ciphertext is computationally distinguishable with respect to two candidate messages.

In knapsack based ECC approach (Rajaram et al 2009) message to be encrypted was converted into ASCII value, and then it was given as input to ECC algorithm. In order to provide better security, knapsack algorithm was applied in encrypted message and then transmitted to receiver. In receiver side, reverse knapsack process and decryption was done to recover the original message. It was combination of encoding and decoding process in knapsack and reverse knapsack algorithm respectively.

Proposed Method

The proposed scheme is divided into three phases: Initial Phase, Signature Generation and Message Recovery.

Initial Phase

During initial phase, server selects domain parameters and all clients compute the public and private keys. Server's database contains information about clients such as their ID and password. Clients transmit their ID and password to server for authentication. Server checks its database to verify whether the client has given the proper password for the corresponding ID, then the server accepts the client as an authenticated client.

Domain parameters for elliptic curve over *Fp* are *p, a, b, Gm* and *n. G* is the generator point (x_G, y_G), a point on the elliptic curve chosen for cryptographic operations and '*n*' denotes order of the elliptic curve. Standards for Efficient Cryptography (SEC), provides predefined values for domain parameters *{*112,128,160,192,224,256,384,521 bits*}* which have a standard curve. In this approach, 128 bit prime field is considered and then corresponding values for *p, a, b, n, G* are chosen from SEC. After server selects the domain parameters, it is sent to all authenticated clients who compute private and public keys with help of domain parameters. Clients selects a random number '*x*' ε [1, p-1] where *x* is private key, then computes public key using

$$y = x.G\ [G+G+\ldots. + x \text{ times}] \quad (1)$$

Public key of all authenticated clients is sent to server. If any client needs other client's public keys, then request should be sent to server in order to obtain it. For each session communication between any two clients, server generates a pair of private and public key $\{PR_K , PU_K\}$.

Signature Generation

During signature generation phase, if sender wants to sign a message and sends it to the receiver, then the first sender has to select a random number k ε (1, p-1) and converts the message M to ASCII form. The sender computes *r* and *s* by:

$$r = M + x_1 \bmod n \quad (2)$$

$$s = [k - (HASH(r) * \text{private key of sender})] \bmod n \quad (3)$$

Choose another random *k* if *r* =0 in equation (2). Apply SHA-256 algorithm to *r* value in order to produce the hash value. The inputs to knapsack algorithm are *r* and *s* along with series. In 1970, Merkle and Hellman inverted the knapsack algorithm which is a public key

cryptography algorithm. It encodes and decodes the given messages. The series used in knapsack algorithm should be super increasing sequence in which the next term of the sequence is greater than the sum of all preceding terms. It is easy to solve a super increasing knapsack by considering the total weight of the knapsack and comparing it with the largest weight in the sequence. If the total weight is less than the largest weight, then it is not in the knapsack, otherwise it is in the knapsack. Subtract the number from the total, and compare with the next highest number. This methodology is continued until the total reaches zero, if not, then there is no solution.

In 1858, Stern defined the Stern sequence as follows:

Stern (1) =1

Stern (n) = Stern (n/2)) if n is even

Stern (n) = Stern (n-1)/2)+Stern((n+1)/2) if n is odd (5)

In Stern series, each row is created by inserting the sum of pair of consecutive elements into the previous row. Stern showed that $gcd(s(n),s(n+1))=1$ and that for every pair of relatively prime positive integers (a,b) there exists in a unique $n>=1$ with $s(n)=a$ and $s(n+1)=b$. When $(a,b)=(0,1)$, it is easy to observe that each row of the diatomic array repeats as the first half of the next row down. Stern series is modified according to the super increasing sequence constraint. The r and s values are converted into binary form which acts as input to knapsack algorithm along with modified Stern series. The output of knapsack algorithm provides encoded r and s value.

The encoded r and s value are double encrypted with the first layer of encryption using receiver's public key and the second layer of encryption using server's public key which ensures confidentiality. ECC is used for encryption process and the steps involved in ECC encryption algorithm are: the first step is to transform the input into points.

In Koblitz's method (1987) to encode input to points, select parameter 'k' which is a random number. Then for each input 'm', compute $x=mk+1$ and solve the corresponding y value ($y^2=x^3+ax+b$). In the second step, select a random number that lies in the range of 0 to n -1. ECC algorithm transforms the input to points. In the third step, compute the ciphertext as {kG, P_{ml}+k* public key of receiver} where P_{ml} denotes (x, y) coordinates which is output of Koblitz's method. Now the double encrypted signature is sent to server.

Message Recovery

When the signature reaches the server, it is decrypted using server's private key. ECC algorithm is used for decryption process. The steps involved in ECC decryption algorithm are: the first step, manipulate the first point (kG) with private key of receiver where kG be the first point and P_{ml} + k*P_B be the second point. The second step involves computation of P_{ml} using

P_{ml} + kP_B- n_BkG

$\Rightarrow$ P_{ml} + k(n_BG)- n_BkG (6)

The third step involves decoding of point back to input using Koblitz's method where for each point (x, y), compute $m= (x-1)/k$ for decoding the point (x, y) to back the symbol 'm'.

Then encrypt the signature using the secret key which is generated based on elliptic curve diffe-Hellman, shared between server and receiver. In the elliptic curve Diffe-Hellman (ECDH) key agreement, the two communicating parties agree beforehand to use the same curve parameters and base point G. They generate their private keys Sa and Ca, respectively, and the corresponding public keys Sb = Sa × G and Cb = $Ca \times G$. Both the client and server exchange their public keys, and each multiplies its private key with the other party's public key to derive a common shared secret key $Sb \times Ca = Ka = Sa \times Cb$. An attacker cannot determine the secret key. The encrypted signature is sent to receiver.

The receiver first decrypt the signature using secret key generated with help of elliptic curve Diffie-Hellman algorithm. Then it again decrypt the signature using receiver's private key to obtain R and S value. Reverse knapsack algorithm is used in order to obtain the signature (r, s) from R and S value using formula R-n^m in an iterative fashion. If $R- n^m > 0$, then assign binary bit 1 at the m^{th} position. The current value is $R=R-n^m$. If value is negative then assign binary bit 0 and R value remains the same. Subtract n^{m-1} from the current R. Depending upon whether it is $+ve$ or $-ve$; assign 1 or 0 at the relevant bit position. Then continue this process until the series becomes null. This will recover the binary bit pattern of r and s value. Then convert binary form of r and s to integer form. Receiver recovers message M using

M=r-[sG+H(r)(public key of sender)$_x$] mod n (7)

Proof

M = r-[sG+H(r)(publickey of sender)$_x$] mod n

$=M+(kG)_x-[(k-H(r)\times$ sender'sprivatekey) G + $H(r)$(public key of sender) $_x$ mod n]

$= M + (kG)_x - (kG)_x + H(r) \times$ sender's privatekey $\times$ G- H(r) public key of sender mod n

⇨ Message

Security Analysis

There are number of attacks against the proposed scheme and the two basic attacks against public-key digital signature schemes are key-only attacks and message attacks. In key-only attacks, an intruder knows only the signer's public key and attempts to derive the sender's private key from known domain parameters (E, p,n and G point, public key of sender) which is not possible in our proposed scheme as the intruder cannot derive $y = x$ and it is difficult to obtain sender's private key due to ECDLP. In message attacks, intruder attempts to forge a digital signature to impersonate sender, which is not possible in the proposed scheme as the knapsack series along with sender's private key has to be known.

Experimental Results

In existing system, knapsack algorithm uses exponential series which may degrade the systems performance. In proposed system, use of modified Stern series in knapsack algorithm reduces the time complexity effectively. Stern series provides better security when compared to exponential series because if any value in series is known to intruders, successive terms can be easily found, which is impossible in Stern series. There is also reduction in time complexity while using Stern series. The computational time for different authenticated encryption schemes is shown in the Table 1.

T_{ECmul} is used to indicate the time to multiply a number by a point on the elliptic curve. T_{ECadd} is the time for the adding one point to another on the elliptic curve. T_{mul} is the time for multiplication. T_h is the time to execute hash function. T_{exp} is the time for exponentiation with mod$P$. T_{inv} is the time for inversion mod$P$. T_{KV} is the time for knapsack value generation. T_{inKV} is the time for inverse knapsack value generation. Chen et al. scheme, requires the computational cost for signature generation phase of $2T_{ECmul}+T_{ECadd}+T_{mul}+T_h$. In the Hsu and Wu scheme, the signer generates a signature that the computational cost is $3T_{exp} +T_{mul}$.

TABLE 1. COMPUTATIONAL TIME FOR DIFFERENT AUTHENTICATED ENCRYPTION SCHEMES

Schemes	Computational Time			
	Signature Generation phase		Signature Verification Phase	
Chen et al [1998]	$2T_{ECmul} + T_{ECadd} + T_{mul}+ T_h$	91.825776	$\mathbf{3}T_{ECmul} + \mathbf{2}T_{ECadd} + T_h$	134.771934
Hsu-Wu [2002]	$3T_{exp} + T_{mul}$	7.947362	$3Texp + (2t + 1)Tmul+ (t\text{-}1)Tinv$	12.474339
Wu-Lin [2008]	$5T_h + 2T_{inv} + 3T_{mul}+ T_{ECadd} + 4T_{ECmul}$	194.79795	$5Th + Tmul+ 2T_{ECadd} + 5T_{ECmul}$	230.974826
Knapsack based ECC [2011]	$T_{ECmul} + T_{mul} + T_h+ T_{KV}$	48.658936	$T_{ECmul} + T_{ECadd} + Th+ TinKV$	47.356199

TABLE 2. COMPARISON BETWEEN EXPONENTIAL AND STERN SERIES FOR SIGNATURE GENERATION

Message	Execution time for Signature Generation	
	Existing series (nanoseconds)	Modified Stern series (nanoseconds)
!@#$%^&*()	25048714	16091501
01234567891011	40912673	32714841
PASSWORD SSN COLLEGE	38122234	20406853
PASSWORD PIN NUMBER 0234	35376912	14268294
MOBILE SMS BANKING APPLICATION	46742673	14947851
KNAPSACK BASED ECC USING STERN SERIES FOR DIGITAL SIGNATURE AUTHENTICATION	346112075	22143037

Knapsack based ECC requires $T_{ECmul} + T_{mul} + T_h+ T_{KV}$. For comparison between existing and proposed systems, consider the message with size of 128 bytes and analyze the time needed to sign and verify as shown in the Table 2 and 3. In exponential series, n can be random number which is less than p bits which is used in modulo arithmetic or $n= pk$ where k is random integer.

Table 2 shows the execution time for signature generation in both exponential and Stern series. Table 3 shows the execution time for message recovery in both exponential and Stern series. Execution time for Stern series is less when compared to exponential series.

Application in Mobile Banking

The algorithm proposed is implemented in mobile banking. In traditional system to obtain mobile

banking service, customers have to go to the bank to submit the registration form by giving their mobile number, account number and transaction details. Then each of them is given a 4-digit number (ATM pin number) for authentication through postal communication which may not be confidential.

TABLE 3. COMPARISON BETWEEN EXPONENTIAL AND STERN SERIES FOR MESSAGE RECOVERY

Message	Execution time for message recovery	
	Existing series (nanoseconds)	Modified Stern series (nanoseconds)
!@#$%^&*()	73572899	80678131
01234567891011	77052182	57639728
PASSWORD SSN COLLEGE	123881177	57877328
PASSWORD PIN NUMBER 0234	82294468	63580916
MOBILE SMS BANKING APPLICATION	89324761	69236873
KNAPSACK BASED ECC USING STERN SERIES FOR DIGITAL SIGNATURE AUTHENTICATION	353551499	61230198

After an analysis of the traditional banking services (Ranbir Soram 2009), a system that would provide better security is proposed. This system is called the ECC banking module which receives the text messages from the clients/banks has been made and then send the output back to the banks/users as required after processed. This ECC banking module provides secure data encryption and decryption using public key cryptography. Instead of obtaining the PIN number/password through post, with help of ECC banking module, it is possible to securely transmit it as shown in figure 1. A digital signature using ECC technology used in each module provides message authentication, message integrity and non-repudiation. However, mobile SMS banking in India does not provide these advantages. Mobile banking in android operating system has been implemented successfully with ECC banking module.

1）Implementation of Mobile Banking in Android

Bank server has the ATM pin number for all customers along with corresponding customer's Unique Identification Number (UID) which is a recently finalized initiative by the Government of India to create and manage a centralized identification system for all the adult citizens and residents of India, for a variety of identification purposes. Bank server will send an alert to customer that he/she would receive pin number within few minutes. When the bank sends the pin number, at that time if the mobile is possessed by the third party other than the desired customer, pin number will be known to third party. In order to avoid that, an alert is sent to customer along with request of customer UID.

After the customer receives that alert, bank will transmit the corresponding UID of the customer. Bank server would check whether the customer has sent the correct UID and then bank generates the public key based on ECC which is sent to customer. When the customer receives the public key from bank, they generate their public key and transmit to bank. Once the bank receives the public key of customer, they generate signature with pin number as input. Then the signature is encoded using knapsack algorithm. Double encryption is then performed on the output of the knapsack algorithm. The first layer of encryption is done using customer's public key and the second layer of encryption is done using secret key based on elliptic curve diffe-Hellman. The double layer of encrypted signature (pin number) is then sent to the customer securely.

ECC banking module has to be embedded in bank server and in the customer handset. When the customer receives the signature, ECC banking module would perform double decryption using secret key and private key. Then the decrypted signature is given to reverse knapsack algorithm in order to decode it. After decoding, PIN number is recovered from the signature with the highest level of security and displayed in the customer's handset.

Conclusions

In this paper, generation of digital signature based on ECC using modified Stern series has been proposed which provides confidentiality, authentication and non repudiation. The time complexity has been reduced drastically while using modified Stern series in knapsack algorithm, when compared to the existing exponential series (Rajaram 2011). In mobile banking, secure transmission of ATM pin number with ECC banking module embedded in bank server and in customer handset has been implemented in Android.

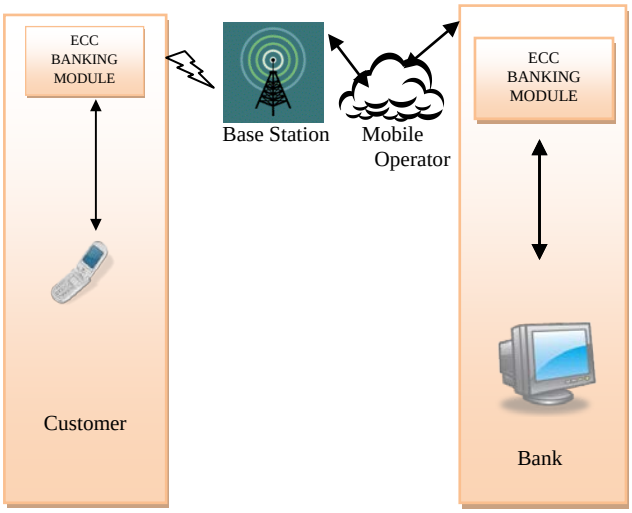

FIG. 1 MOBILE BANKING IN ANDROID

After the customer receives that alert, bank will transmit the corresponding UID of the customer. Bank server would check whether the customer has sent the correct UID and then bank generates the public key based on ECC which is sent to customer. When the customer receives the public key from bank, they generate their public key and transmit to bank. Once the bank receives the public key of customer, they generate signature with pin number as input. Then the signature is encoded using knapsack algorithm. Double encryption is then performed on the output of the knapsack algorithm. The first layer of encryption is done using customer's public key and the second layer of encryption is done using secret key based on elliptic curve diffe-Hellman. The double layer of encrypted signature (pin number) is then sent to the customer securely.

ECC banking module has to be embedded in bank server and in the customer handset. When the customer receives the signature, ECC banking module would perform double decryption using secret key and private key. Then the decrypted signature is given to reverse knapsack algorithm in order to decode it. After decoding, PIN number is recovered from the signature with the highest level of security and displayed in the customer's handset.

REFERENCES

Chen, T. S., Huang, G. S., Liu, T. P., and Chung, Y. F., "Digital signature scheme resulted from identification protocol for elliptic curve cryptosystem", Proceedings of IEEE TENCON'02, (2002): 192-195.

ElGamal, T., "A public-key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, 31 (1985): 469-472.

Hsu, C.L. and Wu, T. C., "Authenticated encryption scheme with (t, n) shared verification", IEEE Proceedings - Computers and Digital Techniques, 145 (1998): 117-120.

Kadhm, Noor Dhia, "Construction of a hard direct digital signature scheme", Journal of Kerbela University, 10 (2012): 77-84.

Koblitz, N., Elliptic curve cryptosystems, Mathematics of Computation, 48 (1987): 203-209.

National Institute of Standards and Technology (NIST), "The digital signature standard proposed by NIST", Communications of the ACM, 35 (1992): 34-40.

Nyberg, K. and Rueppel, R. A., "A new signature scheme based on the DSA giving message recovery", ACM Computer and Communications Security, 1 (1993): 58-61.

Popescu, C., "An Identification Scheme Based on the Elliptic Curve Discrete Logarithm Problem," Proceedings of the Fourth International Conference/ Exhibition on High Performance Computing in the Asia-Pacific Region, (2000) Vol. : 624-625.

Ramasamy, R. Rajaram and Prabakar, M. Amutha," Digital Signature Scheme with Message Recovery Using Knapsack-based ECC", International journal of Network security, 12 (2011): 15-20.

Ramasamy, R. Rajaram, Prabakar, M. A., Devi, M. I., and Suguna, M., "Knapsack based ECC encryption and decryption", International Journal of Network Security, .9 (2009): 218-226.

Rivest, R. L., Shamir, A., and Adleman, L., "A method for obtaining digital signatures and public key cryptosystems", Communications of the ACM, 21 91978): 120-126.

Sonali, N, Malk, L.G.,"Prospective utilization of ECC for security enhancement" International Journal of Application or Innovation in Engineering & Management, 2 (2013).: 87-92

Soram, Ranbir, "Mobile SMS Banking Security Using Elliptic Curve CryptoSystem" International Journal of Computer Science and Network Security, 9 (2009): 30-38.

Subramanian, Asvhini, M.Ahamed, Chaudhri, ,"A study on elliptic curve digital signature algorithm for reliable e-commerce applications", Smart computing review, 2 (2012): 71- 78

Wu, T. S. and Lin, H. Y., "ECC based convertible authenticated encryption scheme using self-certified public key systems", International Journal of Algebra, 2 (2008): 109-117.

Latha Parthiban is with Pondicherry University in the Department of Computer Science. She has B.E degree in Electronics and Communication Engineering from Madras University, M.E in Computer Science and Engineering from Anna University, Chennai and PhD in Computer Science and Engineering from Pondicherry University. Her area of interest includes data mining, network security and image processing.

Nivethaa Sree is with the Department of Information Technology of Sri Venkateswara College of Engineering, Sriperumbutur, Chennai. She has B.E in Computer Science and Engineering and M.E in Computer Science and Engineering from Anna University, Chennai.